Exploration of Central Limit Theorem

Ben Kesseler

June 19, 2016

Overview

I will create 1000 simulations of a data set with 40 data points from an exponential distribution. In addition, I will explore the mean and variance of the means of those 1000 data sets, in comparison with their theoretical values.

Simulations

First I need to create the data for my simulation. I'll create a matrix with 1000 rows, 40 columns, where each cell represents a pull from an exponential distribution.

```
library(ggplot2)
```

```
lambda <- 0.2 # Setting the rate for the exponential distribution
n <- 40 # Setting the number of samples to take, per average
simulations <- 1000 # Setting the number of total simulations

# Set Seed for Reproducibility
set.seed(314152)

# Creating the overall simulations
matrix_simulations <- matrix(data = rexp(n*simulations, rate = lambda),
                             nrow = simulations)

# Creating the averages
simulation_means <- data.frame(means = apply(matrix_simulations, 1, mean))
```

Sample Mean versus Theoretical Mean

The theoretical mean should be 1/lambda = 5. I'll now plot the distribution, including the sample mean and the theoretical mean, along with the difference.

```
annotate1 <- paste("Black Line = Sampled Mean =",
                   round(mean(simulation_means$means), 3))
annotate2 <- paste("Red Line = Theoretical Mean =", 1/lambda)
annotate3 <- paste("Difference from Theory =",
                   round(mean(simulation_means$means) - 1/lambda, 3))

ggplot(data = simulation_means, aes(x = means)) +
        geom_histogram(aes(y = ..density..),
                       binwidth = 0.1,
                       col = "black",
                       fill = "white") +
        geom_density(col = "blue",
                     size = 2,
                     fill = "blue",
                     alpha = 0.3) +
        theme_bw() +
        ggtitle("Distribution of 1000 Simulations of the Mean of 40 Exponential Data Points") +
        geom_vline(xintercept = mean(simulation_means$means),
                   size = 1,
```

```
                col = "black") +
      geom_vline(xintercept = 1/lambda,
                 size = 1,
                 col = "red") +
      annotate("text", x = 6, y = 0.5, label = annotate1, hjust = 0) +
      annotate("text", x = 6, y = 0.45, label = annotate2, hjust = 0) +
      annotate("text", x = 6, y = 0.4, label = annotate3, hjust = 0)
```

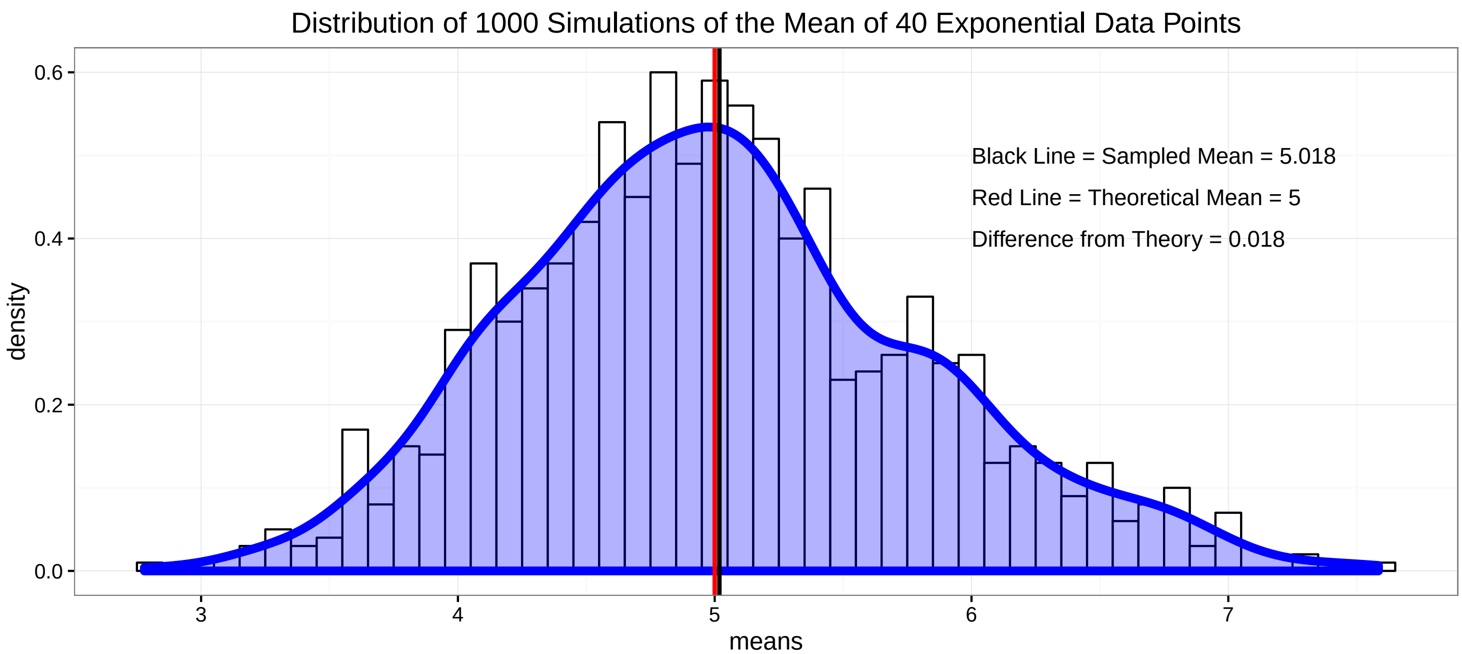

Figure 1: The plot shows the probability distribution of my simulations, along with the theoretical mean.

So, the mean I calculate is close to the theoretical mean, with a difference of only 0.0184027.

Sample Variance versus Theoretical Variance

The theoretical variance is 1/lambda2 = 25. I'll now calculate the sample variance, and the difference from the theoretical variance.

```
# Theoretical Variance
theoretical_variance <- (1/(lambda*sqrt(n)))^2
theoretical_variance
```

```
## [1] 0.625
```

```
# Sample Variance
sample_variance <- var(simulation_means$means)
sample_variance
```

```
## [1] 0.624341
```

```
# Difference in Variance
variance_difference <- sample_variance - theoretical_variance
variance_difference
```

```
## [1] -0.0006590431
```

So the sample variance is quite close to the theoretical variance, which is as expected.

Distribution

Now I will plot the distribution of my samples, along with the normal distribution I would expect to see, which would be a normal distribution with the mean equal to 1/lambda = 5 and the standard deviation equal to 1/(lambda * sqrt(n)) = 0.7905694.

```
annotate4 <- paste("Blue Line = Sampled Distribution")
annotate5 <- paste("Red Line = Theoretical Distribution")

ggplot(data = simulation_means, aes(x = means)) +
        geom_histogram(aes(y = ..density..),
                       binwidth = 0.1,
                       col = "black",
                       fill = "white") +
        geom_density(col = "blue",
                     size = 2,
                     fill = "blue",
                     alpha = 0.3) +
        stat_function(fun = dnorm,
                      args = list(mean = 1/lambda, sd = sqrt(theoretical_variance)),
                      colour = "red", size = 2) +
        theme_bw() +
        ggtitle("Distribution Comparison - Theoretical vs Sampled (1000 Simulations)") +
        annotate("text", x = 6, y = 0.5, label = annotate4, hjust = 0) +
        annotate("text", x = 6, y = 0.45, label = annotate5, hjust = 0)
```

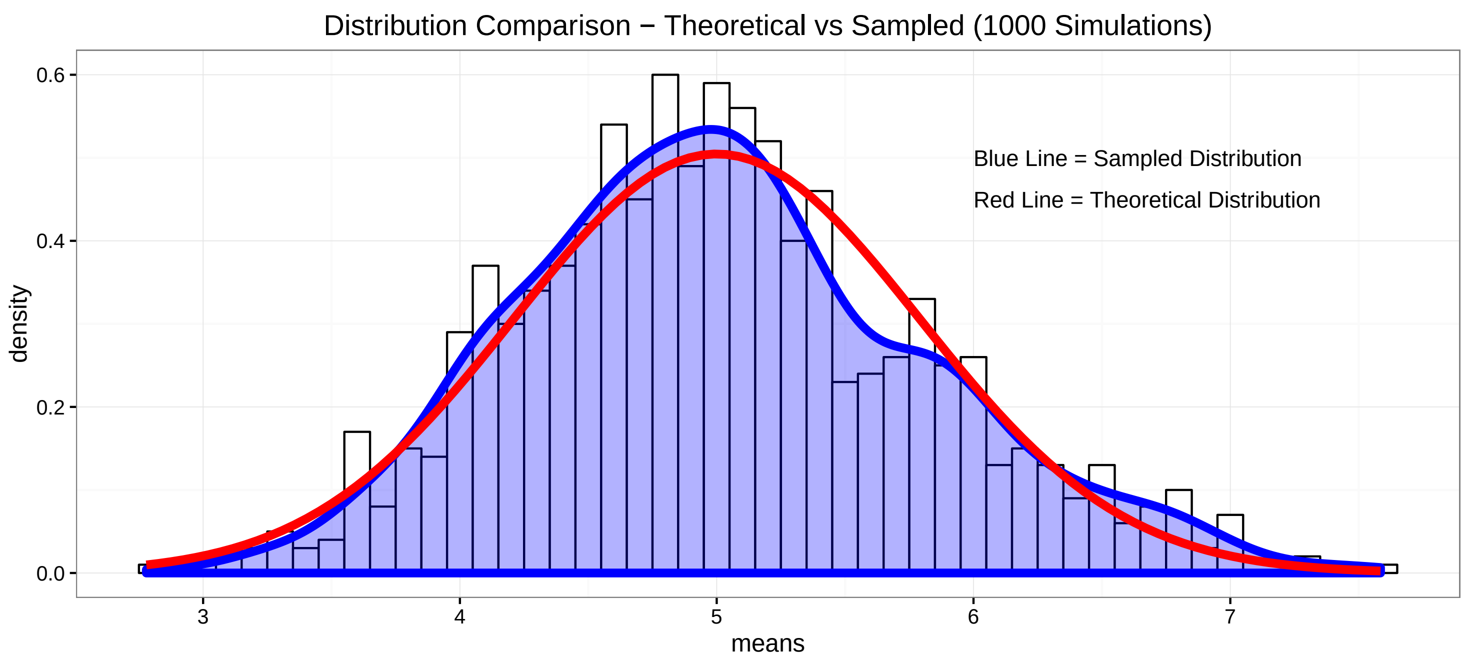

Figure 2: Comparison of Sampled Distribution and Expected Theoretical Normal Distribution, 1000 simulations

As I expected, they are close, albeit not perfectly aligned.

Appendices

100k Simulations!

Let's try it with 100k simulations, instead of 1000. Maybe the final distribution would look closer to a normal even still!

```
lambda <- 0.2 # Setting the rate for the exponential distribution
n <- 40 # Setting the number of samples to take, per average
simulations <- 100000 # Setting the number of total simulations

# Creating the overall simulations
matrix_simulations_100000 <- matrix(data = rexp(n*simulations, rate = lambda),
                                    nrow = simulations)

# Creating the averages
simulation_means_100000 <- data.frame(means = apply(matrix_simulations_100000, 1, mean))
```

```
ggplot(data = simulation_means_100000, aes(x = means)) +
        geom_histogram(aes(y = ..density..),
                       binwidth = 0.1,
                       col = "black",
                       fill = "white") +
        geom_density(col = "blue",
                     size = 2,
                     fill = "blue",
                     alpha = 0.3) +
        stat_function(fun = dnorm,
                      args = list(mean = 1/lambda, sd = sqrt(theoretical_variance)),
                      colour = "red", size = 2) +
        theme_bw() +
        ggtitle("Distribution Comparison - Theoretical vs Sampled (100000 Simulations)") +
        annotate("text", x = 6, y = 0.5, label = annotate4, hjust = 0) +
        annotate("text", x = 6, y = 0.45, label = annotate5, hjust = 0)
```

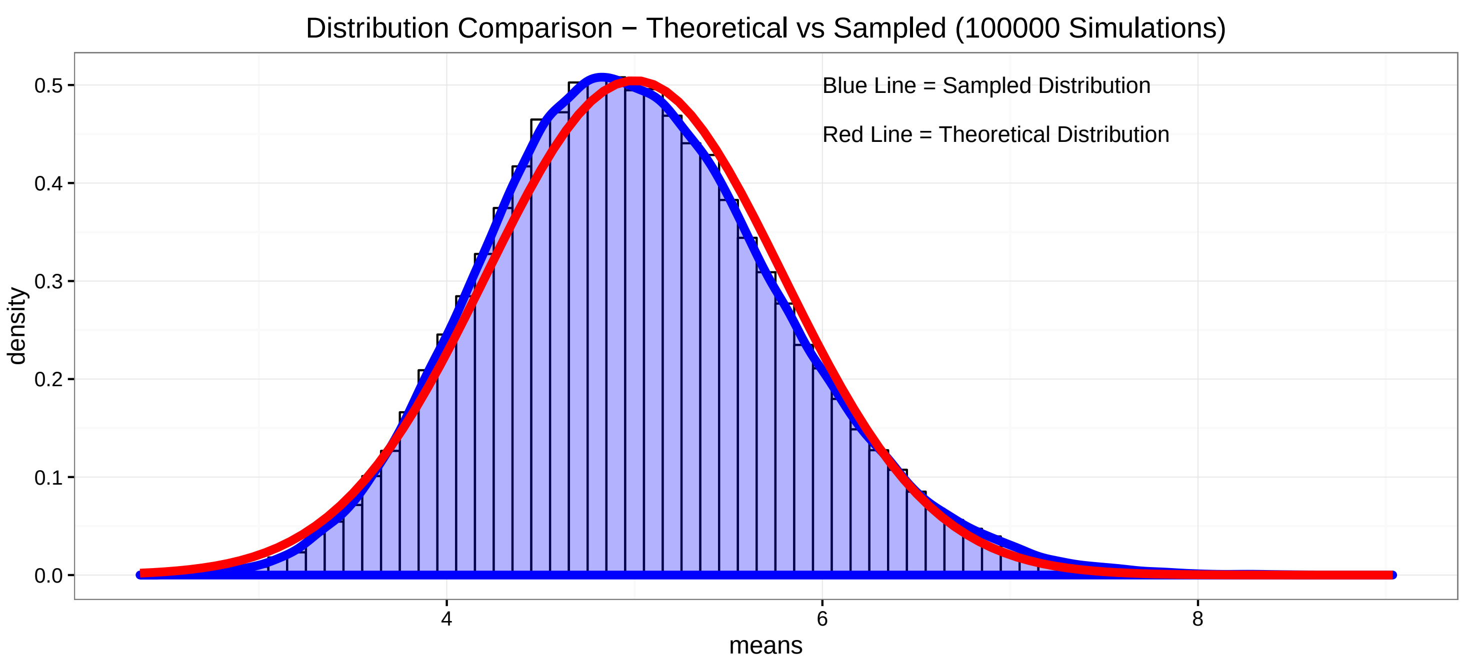

Figure 3: Comparison of Sampled Distribution and Expected Theoretical Normal Distribution, 100000 simulations

Much closer!

100k Simulations with 100 data points per simulation!

Let's try it with 100k simulations, instead of 1000. Maybe the final distribution would look closer to a normal even still!

```
lambda <- 0.2 # Setting the rate for the exponential distribution
n <- 100 # Setting the number of samples to take, per average
simulations <- 100000 # Setting the number of total simulations

# Creating the overall simulations
matrix_simulations_100000 <- matrix(data = rexp(n*simulations, rate = lambda),
                                    nrow = simulations)

# Creating the averages
simulation_means_100000 <- data.frame(means = apply(matrix_simulations_100000, 1, mean))
```

```
theoretical_variance <- (1/(lambda*sqrt(n)))^2

ggplot(data = simulation_means_100000, aes(x = means)) +
        geom_histogram(aes(y = ..density..),
                       binwidth = 0.1,
                       col = "black",
                       fill = "white") +
        geom_density(col = "blue",
                     size = 2,
                     fill = "blue",
                     alpha = 0.3) +
        stat_function(fun = dnorm,
                      args = list(mean = 1/lambda, sd = sqrt(theoretical_variance)),
                      colour = "red", size = 2) +
        theme_bw() +
        ggtitle("Distribution Comparison - Theoretical vs Sampled (100000 Simulations, 100pts/Simulation)") +
        annotate("text", x = 6, y = 0.5, label = annotate4, hjust = 0) +
        annotate("text", x = 6, y = 0.45, label = annotate5, hjust = 0)
```

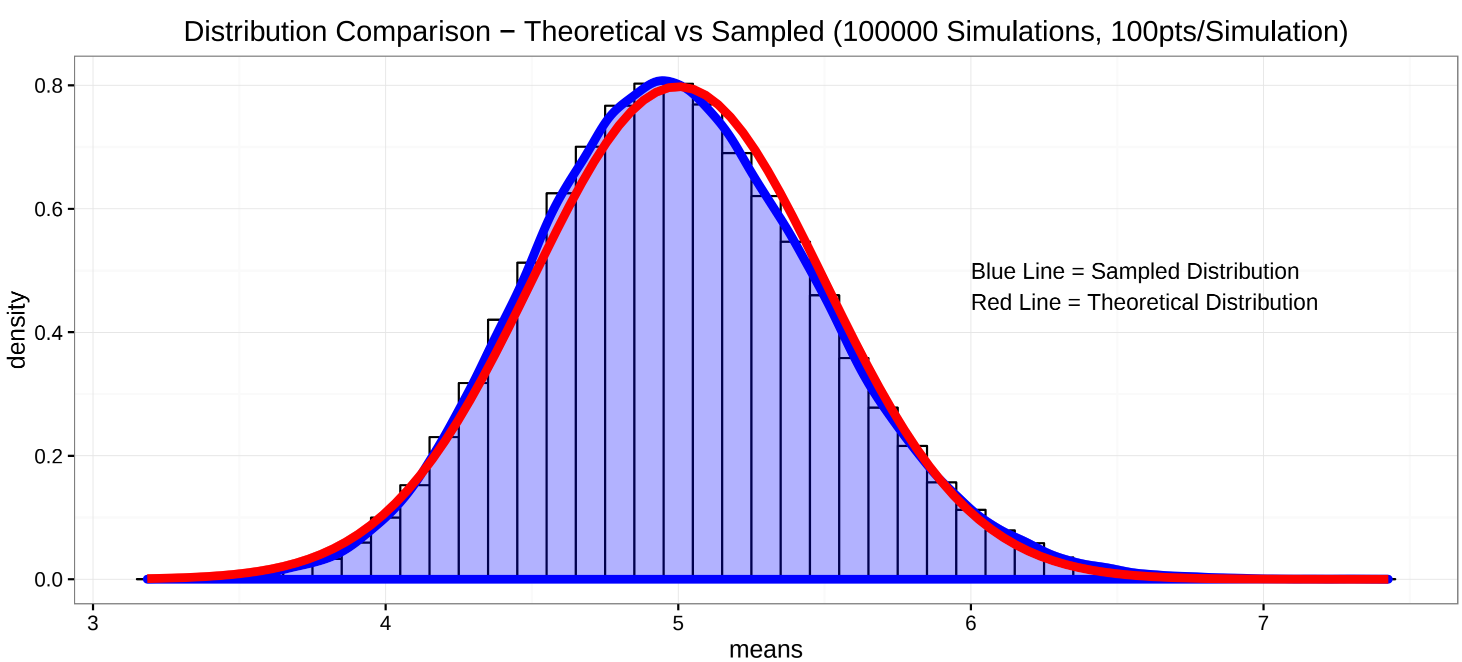

Figure 4: Comparison of Sampled Distribution and Expected Theoretical Normal Distribution, 100000 simulations, 100 data points per simulation

Much closer still!

Additional Resources

- Github Repository

Session Information

- 3.20 GHz Intel i5 650
- 8GB RAM
- RStudio Version 0.99.902

```r
sessionInfo()
```

```
## R version 3.3.0 (2016-05-03)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 10586)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggplot2_2.1.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.5      digest_0.6.9     plyr_1.8.4       grid_3.3.0
##  [5] gtable_0.2.0     formatR_1.4      magrittr_1.5     evaluate_0.9
##  [9] scales_0.4.0     stringi_1.1.1    rmarkdown_0.9.6  labeling_0.3
## [13] tools_3.3.0      stringr_1.0.0    munsell_0.4.3    yaml_2.1.13
## [17] colorspace_1.2-6 htmltools_0.3.5  knitr_1.13
```